Annex 3 – Unlimited Size Output

A3.1 - Why do current LLMs have an output window limitation?

LLMs limit their output window because the computational cost grows quadratically with the number of generated tokens, and beyond a certain point, this becomes unfeasible in terms of both time and memory.

In current Transformer-based architectures, each newly generated token must compute its attention over all previously generated tokens. This means that at step ***t***, the model performs an attention operation involving the previous ***t–1*** tokens. As the sequence grows this computation becomes progressively more expensive, since attention is performed cumulatively rather than at a constant rate.

The total computational cost to generate a sequence of ***N*** tokens in a modern Transformer model is formally expressed as:

$$C_{total}(N) = \sum_{t=1}^{N} C_{token}(t) = L \,.\, H \,.\, d_k \sum_{t=1}^{N} t = L \,.\, H \,.\, d_k \, \frac{N\,(N+1)}{2} \approx \frac{L \,.\, H \,.\, d_k}{2} N^2$$

Where:

L: number of layers in the Transformer model.

H: number of attention heads per layer.

d_k: internal dimension of each attention head.

N: total number of output tokens.

$C_{token}(t)$: cost of generating the token at position ***t***.

Simplified formula:

In practice, and to emphasize the quadratic growth with respect to the number of output tokens, the formula can be simply expressed as:

$$C_{total}(N) \;\alpha\; N^2$$

This expression means that the total cost grows quadratically with ***N***.

In other words: doubling the output length quadruples the computational cost. For this reason, the output windows in LLMs are strictly limited.

In summary:

The quadratic growth of computational cost, together with the physical limitations of memory and hardware resources, makes it unfeasible to generate long outputs in a single pass.

The quadratic cost N^2 is so restrictive that, beyond certain values of ***N***, the process becomes unviable even in advanced infrastructures, which explains why no commercial or open source model allows unlimited outputs in a single pass.

Moreover, as the sequence grows, the probability of cumulative errors (drift) also increases, affecting the coherence and accuracy of the output. **All of this justifies the need for alternative paradigms.**

A3.2 - How does a human overcome their own limitations?

It is striking that a writer, with a brain running on just 20 watts of power can write entire books, while a modern AI consuming megawatts of power is still unable to accomplish such tasks. How does the human brain solve this problem? In the following way….

Practical analogy: The student in the library

Let us imagine a student who enters a library to carry out an extensive practical assignment, needing to write a lengthy essay. This student does not produce the entire text in a single attempt. Instead, they follow an organized, structured and deliberate process:

Step 1: Generation of the conceptual index

First, the student writes the "table of contents" for the intended answer or exposition. This forms an index that acts as a structural skeleton.

Step 2: Development by fragments

For each point in the index, the student writes the corresponding section. Whether it is a paragraph or a chapter, each fragment is generated independently and stored.

Step 3: Assembly

The stored fragments are then assembled and concatenated according to the developed index.

Step 4: Revision

After all the topics outlined in the planned index have been assembled and concatenated, the student conducts revision phases to ensure the coherence and logical flow of the entire text.

In this way, the student can construct a response or document of any length, easily overcoming any physical or immediate attention limitations.

A3.3 - Solution Extrapolated to an Algorithm According to CC-EI Output Chaining

The solution proposed by the Concept Curve (CC) paradigm is to model the process of unlimited output generation not as a monolithic task, but as a modular and dynamic construction based on conceptual decomposition and semantic indexing.

> ***Clarification****: The generation and assembly process described below, according to the Concept Curve paradigm,* ***does not require the use of vector embeddings or Retrieval-Augmented Generation (RAG) techniques****. Both the conceptual index and the fragments ("chunks") are generated and organized explicitly and sequentially, without involving semantic search or vector indexing processes. In other words:* ***(1)*** *The AI is fully capable of creating the index of a document, and* ***(2)*** *the AI is fully capable of writing the content for each section of that index. At no point does it need to compress, compare, or decompress vectors.*

Solution expressed as an Algorithm:

Step 1: Generation of the conceptual output index

Before generating the final response, the system creates an index of key concepts that the output should cover. This index, according to the Concept Curve paradigm, acts as a guiding map of topics, subtopics, and the logical sequence of the expected content.

Step 2: Output chunking

For each concept or group of concepts in the index, partial or independent "chunks" of responses are generated, each of which addresses a specific part of the output, and are stored temporarily.

Step 3: Assembly – narrative and conceptual merging

Once all the chunks have been generated, they are sequentially combined according to the Concept Curve indexing.

Step 4: Revision and iterative output

The indexed and modular nature of CC-EI[1] allows any fragment that is insufficient or ambiguous to be regenerated or expanded at any time, without the need to regenerate the entire document.

In summary, this approach solves the output limitation problem not through brute force, but through **planning and modular assembly**.

[1] CC-EI Concept Curve Embeddings Indexation

A3.4 – Conclusion

Advantages of approaching the solution through the Concept Curve paradigm:

- There is no theoretical limit to the output length, since generation and assembly by fragments make it possible to create documents of any length.
- The response remains coherent and faithful to the conceptual structure defined at the outset.
- Computational efficiency is maximized, as only what is strictly necessary is generated, avoiding redundant processing.
- It is possible to audit and modify the output at any stage, increasing transparency and control.

In summary:

By using the Concept Curve paradigm, the restriction on output length is overcome by dividing the generation process into stages: (1) creation of the conceptual index, (2) production of fragments (chunks), (3) assembly, and (4) revision. This approach surpasses the inherent limitations of traditional language models and paves the way for advanced applications in knowledge generation, long-form texts, and automated reasoning.

All of this is achieved while minimizing computational cost.